Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 73 – 82

Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Construction of data aggregation tree for multi-objectives in wireless sensor networks through jump particle swarm optimization

Yao Lu[a,b,*], Jianping Chen[b], Ioan Comsa[b], Pierre Kuonen[b], Beat Hirsbrunner[a]

[a]*Department of Informatics,University of Fribourg, CH-1700, Switzerland*

[b]*Institute of Complex Systems,University of Applied Sciences of Western Switzerland, CH-1705, Switzerland*

Abstract

As a typical data aggregation technique in wireless sensor networks, the spanning tree has the ability of reducing the data redundancy and therefore decreasing the energy consumption. However, the tree construction normally ignores some other practical application requirements, such as network lifetime, convergence time and communication interference. In this case, the way how to design a tree structure subjected to multi-objectives becomes a crucial task, which is called as multi-objective steiner tree problem (MOSTP). In view of this kind of situation, a multi-objective optimization framework is proposed, and a heuristic algorithm based on jump particle swarm optimization (JPSO) with a specific double layer encoding scheme is introduced to discover Pareto optimal solution. Furthermore, the simulation results validate the feasibility and high efficiency of the novel approach by comparison with other approaches.

Peer-review under responsibility of KES International.

Keywords: Spanning Tree; Data Aggregation; Wireless Sensor Networks; Steiner Tree; Multi-objective Optimization, PSO

1. Introduction

One of the most significant functions of wireless sensor networks (WSNs) is gathering data from the environment for a long duration[1]. Since sensors are intentionally and densely deployed, the data gathering events are possible to concurrently trigger the responding actions from a portion of sensors. In the normal case, direct data transmission from source nodes to the sink node leads to high data redundancy and communication load. Therefore, data aggregation is developed to address this problem[2], and there are various techniques used as in-network data aggregation[3]. Tree aggregation as a typical technique outperforms others on long-term and static aggregation events. Its general principle is gathering data based on the tree structure, source nodes transmit original data to relaying nodes, which have the aggregation function and are responsible for eliminating the redundant data, and afterwards the aggregated result is transmitted to the higher capable relaying nodes until the sink node is reached. The longer distance communications

* Yao Lu. Tel.: +41-078-972-0856.
E-mail address: yao.lu@unifr.ch

Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.086

are transformed to local communication by taking the advantage of the relaying nodes, and the intolerable energy consumption is also avoided.

Nevertheless, there is a significant issues that has to be considered: which sensors are selected as the relaying nodes. It can be abstracted to an NP-complete combinatorial optimization problems, known as Steiner Tree Problem (STP). Given a weighted graph in which a subset of nodes are identified as terminals (sink and source nodes), find a minimum-weight connected subgraph that includes all the terminals. For the purpose of discovering the efficient implementation of tree aggregation, there are multiple performance metrics to evaluate the structure. The selection of metrics is depending on the concrete system requirements. For instance, energy consumption, convergence time, network lifetime, and communication interference are the most conventional performance criterions. Supposing that multiple metrics are concerned simultaneously, constructing the aggregation tree becomes an multi-objectives optimization problem.

The combinatorial issue of multi-objective optimization and STP is called MOSTP in this paper. An approximate algorithm to find the near-optimal solution in polynomial time for MOSTP represents the main contribution of this paper. Based on definitions and analysis of the above issue, a heuristic method is implemented by modifying JPSO. The special double layer encoding scheme and other necessary components are used for the proposed method development. The rest of the document is organized as follows: Section II presents the elements of the related work. Section III describes the multi-objective optimization problem. Section IV highlights the implementation of the proposed approach. Section V presents the results and finally paper concludes with Section VI.

2. Related works

Based on the existing literature[4], WSN is represented as a weighted topology graph, where the weight is denoted as distance, load or other metrics. To find the efficient structure, the algorithm addressed in[5] constructs the aggregation tree in an energy and latency efficient manner. However, it is applied only on the special case of the general data aggregation problems in which all the nodes in the network are source nodes, and it leads to the generality lost for the optimization problem.

By making a generalization of the optimization model, MST-1tRNP as an approximation algorithm based on minimum spanning tree algorithm is proposed in[6]. It aims to solve the relay placement problem in WSNs, which is also STP, but the objectives are different. The tree structure with lowest value of the total link cost can be generated. M-REST is another approximate algorithm based on genetic algorithm proposed to solve a multi-objective relaying nodes placement problem in[7]. However, the data aggregation function is not considered in the model, and only two objectives are considered. Besides, the encoding scheme potentially generates cycle in the structure and requires specific cycle breaking method to keep the feasibility of solutions.

3. Problem description and definition

In order to analyze the MOSTP approach, there are some reasonable assumptions for the wireless sensor networks. The network topology is considered to be static and connected. The communication link for two nodes is symmetrical, and the transmission distance for each node is restricted. The aggregation function is automatically executed on the intermediate nodes on the spanning tree.

3.1. Network Model and Assumption

The WSN has been modeled as an undirected incomplete graph $G(V, E)$, where V is a finite set of sensors, and which has been uniformly or randomly distributed in the areas of monitor regions, the number $|V|$ is n, meanwhile E represent the links between nodes. The number of source nodes has been assumed as $m(m < n)$, and there is only one sink node. Even if the rest of sensor nodes do not have source data, they can help to relay data and to improve the aggregation performance. These relaying nodes can be considered as steiner nodes $SN(|SN| \leq (n - m))$. The spanning tree is defined as $Tree(V_{m+sn}, E_{m+sn})$.

3.2. Energy Dissipation Model

The classical energy dissipation model for wireless sensor communication shown in[8] is adopted. The both free space and multi-path fading channel models have been considered. The energy consumption for reception and transmission of k bits data over distance d are defined as follows:

$$E_{RX} = kE_e \qquad E_{TX} = \begin{cases} kE_e + kE_{fs}d^2, d < d_o \\ kE_e + kE_{mp}d^4, d >= d_o \end{cases} \tag{1}$$

Where E_e represents the energy consumption of radio dissipation, E_{fs} and E_{mp}, respectively represent the energy consumption of amplifier in the free space and the multi-path fading channel model. The data aggregation occurs when there is new receiving data available on the intermediate node and the energy consumption for data aggregation on the intermediate node can be defined as follows, where E_f indicates average unit fusion cost, and α denotes the aggregation factor, k_i and k_j represent the data from sending node i and receiving node j, respectively.

$$E_{DA} = \alpha(k_i + k_j)E_f \tag{2}$$

3.3. Multi-objective Optimization

Multi-objective optimization is a framework in which multiple objective functions are desired to have equal treatments. The solution to this problem is a set of multiple sub-solutions, which optimizes simultaneously the objectives. In addition, the feasible solutions have to comply with certain constraint conditions. Eventually, the decision is acquired after the optimization process has finished. This problem can be formulated as follows:

$$\begin{array}{c} Min(f_1(x), f_2(x), \cdots f_k(x)) \\ s.t. \varphi_i(x) > 0 (i = 1, 2, 3 \cdots p) \end{array} \tag{3}$$

Where $f_k(x)$, $\varphi_i(x)$ denotes the multiple different objectives and the constraint condition respectively. And k, p is the number of objectives and constraints, respectively, and x is the feasible set of decision.

In majority literatures of aggregation tree[3], authors only considered one or two objectives, for instance, the most common cases are energy consumption and latency. It means other important objectives are neglected. In this paper, through the summary of the most common application requirements, four objectives are selected.

Objective 1 : The total energy consumption of tree should be minimized. Due to the primary constraint for sensors is the limited energy, and then the total energy consumption of spanning tree needs to be minimized. The value of the total energy consumption equals to the dissipation sum of all nodes in the tree, the value can be calculated by the energy dissipation model, where *Cnum* denotes the number of the child nodes.

$$f_1(x) = E(Tree_x) = \sum_{i=1}^{m+|SN|} (E_{TX}^i + \sum_{j=1}^{Cnum_i} (E_{RX}^j + E_{DA}^j)) \tag{4}$$

Objective 2 : The network lifetime of tree should be maximized. Network lifetime represents the period in which the aggregation tree is able to maintain the functionality. The node failure caused by the imbalance of energy consumption is the primary reason for the destruction of tree structure. Supposing the first failure occurs on the relaying nodes, the spanning tree has to be reconstructed globally. Therefore, the lifetime of whole tree can be considered as the minimum nodal lifetime of relaying nodes.

$$\begin{array}{c} L(i) = E_{residual}^i / (E_{TX}^i + \sum_{j=1}^{Cnum_i} (E_{RX}^j + E_{DA}^j)) \\ f_2(x) = 1/L(Tree_x) = 1/\underset{i \in SN}{Min}(L(i)) \end{array} \tag{5}$$

Objective 3 : The convergence time of tree aggregation should be minimized. The latency for data aggregation represents the time required from data transmission by the first source node till the last packet is received by the sink in one round. Supposing a tree has the root node at level 0 and the leaf nodes at level h. On the principle of scheduling rules on MAC layer, the time slot at layer i as the transmitting time unit directly impacts the latency $lt(i-1)$ at layer $i-1$. Its length depends on the maximum size of transmitting data $data_i$ and the transmitting rate $rate_i$. Since each parent has to wait for receiving data from all child nodes, the maximum number of child CN_{i-1} also affect the latency at layer $i-1$. According to the analysis in[5], the optimal bound or lower bound of convergence time can be defined as follows.

$$f_3(x) = \sum_{i=1}^{h} lt(i-1) = \sum_{i=1}^{h} CN_{i-1} * (Max(data_i)/rate_i) \tag{6}$$

Objective 4 : The communication interference of tree aggregation should be minimized. The link interference is defined as the number of nodes affected by its communication. The maximum of link interference is considered as the network's interference. The network is not well constructed when interference is high. Since each sensor can selectively decide nodes to communicate by adjusting its transmission power, meanwhile, the transmission disc is determined. If there is a link e between node x and node y, the length of this link $|x, y|$ is actually the radius of the transmission disc x, and disc y. In order to construct the optimal tree structure, the appearance of high-interference link should be reduced in the tree.

$$\begin{gathered} Cov_e = \{n \in V | n\ is\ covered\ by\ D(x, |x, y|)\ or\ D(y, |y, x|)\} \\ f_4(x) = \underset{e \in tree}{Max} |Cov_e| \end{gathered} \tag{7}$$

Due to the restricted power of sensor, the primary objective is total energy consumption. But without considering the network lifetime, the imbalance of energy consumption leads to premature death of some nodes[9]. And long latency for aggregation leads to the decrease of the number of aggregation operation in one time unit. At last, high interference cause the retransmissions which decreases the network performance. Even if these objectives probably affect each other, they can be trade-off through MOSTP optimization model, and the approximate optimal Pareto solution can be located.

4. Jumping particle swarm optimization for MOSTP

4.1. JPSO

As a variant of discrete particle swarm optimization (DPSO), jumping particle swarm optimization (JPSO) has been developed to deal with the minimum labeling Steiner tree problem (MLSTP) by[10]. Due to the similarity of these problems, it can be used to address MOSTP problem. JPSO interprets the weights of the updating equation as probabilities and each particle has a random behavior during each iteration or acts in a way guided by the effect of an attraction. The movements in a discrete or combinatorial space are jumps from one solution to another. The attraction causes the given particle to move towards this attractor if it results in an improved solution. And the update equation is expressed as follows:

$$x_{i+1} = c_1 x_i \bigoplus c_2 b_i \bigoplus c_3 g_i \tag{8}$$

There are 3 components in equation for updating position, and the last three are called attractors: x_i is the original position, which keeps the current position, b_i is the individual best position, which encourages the self cognition, and g_i is the best position in current swarm, which leads to social learning. Depending on the position of three different attractors, the particle has the direction to move towards for a better location. The particle only can move to one of these attractors during each updating step, and the possibility of movement is determined by the factor c_j, and $\sum c_j = 1$.

4.2. Algorithm overview

For the purpose of solving MOSTP by JPSO, each particle is the solution to represent a tree structure, and the primary difficulty derives from designing efficient encoding scheme and evolutionary operator. There are diverse encoding stratergy for tree-based combinatorial problems, Prufer number, Network Random Keys, Edge Set are the typical schemes[11]. However, when these schemes are applied to approximate problems of STP, satisfactory performance can not be guaranteed. In this case, there are two requirement for encoding scheme. First, the encoding can be evolved on incomplete graph, which means the links are included in graph, only if their length are shorter than wireless transmission distance. Second, the involved nodes of encoding are variable in STP, the encoding can be evolved on not only the complete set of all nodes, but also the different subsets of these nodes. These previous schemes potentially generate infeasible solutions. To enable efficient evolution, the double layer encoding scheme is developed to guarantee the particle flying inside feasible solution space. And the overview of the algorithm is described in the following list.

Algorithm 1 JPSO Overview

Require: $Graph(V, E)$
Ensure: $Tree$
    $Initialize(P_1, P_2 \cdots P_j)$
    while stopping criteria is not met **do**
        for each particle P_j **do**
            R=random();
            if $C_0 < R \leq C_0 + C_1$ **then**
                $flag{=}match(P_j, B_j, layer_1)$;
                $P_j{=}Particle\_Flying(P_j, B_j, flag)$;
            else if $C_0 + C_1 < R$ **then**
                $flag{=}match(P_j, G_s, layer_1)$;
                $P_j{=}Particle\_Flying(P_j, G_s, flag)$;
            end if
            $P_j{=}Particle\_Repair(P_j)$;
            $Fitness\_Evaluation(P_j)$;
            $Individual\_Best(B_j)$;
        end for
        $Global\_Best(G_s)$;
    end while
    $Tree = TCR\_Decoder(G_s)$;

The details of necessary sub-components are demonstrated in the following section. *Particle_Flying* is the evolutionary operator. *Particle_Repair* is used to tailor the tree structure and ensure the feasibility of the particle. *TCR_Decoder* is used to generate or to translate the particle encoding to the tree structure.

4.3. Representation of particle

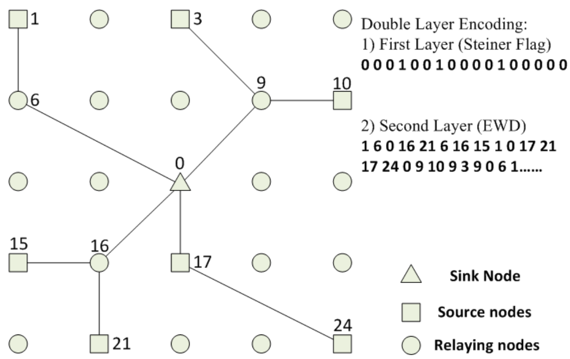

Fig. 1. Spanning tree example

No matter how the tree structure changes, it has to comply with the restriction that the terminals of the spanning tree must be sink and source node respectively, and all source nodes must be included in the tree. In addition, necessary steiner nodes act as relaying nodes to relay and aggregate data. From the view of the tree representation, not only the different steiner nodes correspond to different tree structure, but also the same steiner nodes can match to multiple different tree structure. For the purpose of generating the fine-grained solutions, the double layer encoding scheme is proposed. Supposing the total number of nodes is n, and the number of source nodes is m, there is an example in figure 1.

For the first layer, a binary string is used as steiner flags of the candidate nodes. The second layer is generated based on the content of the first layer and modified into feasible solution. The number of the candidates of steiner nodes is $n - m$, and the identity of these candidates are arranged in ascending order, each binary of encoding indicates whether the correspondent node is selected as the steiner node or not.

$$stFlag(node_i) = \begin{cases} 1, & if\ node_i \in Steiner\ Nodes \\ 0, & otherwise \end{cases} \tag{9}$$

For the second layer, many previous tree encoding strategy attempted to find a way to strike a balance between locality, heritability and feasibility handling in evolutionary algorithms. And the edge window decoder (EWD) representation is proposed to achieve a good balance for this issue, which was proposed by S.M. Soak[11]. It hybridizes a straightforward and direct encoding of a subgraph with a tree-construction algorithm (TCR). Due to the properties of order preserving, node preserving and greedy, the TCR as the decoder interprets inputting single string into an unique spanning tree. An exmaple of TCR decoding process is depicted in figure 2, and two adjoining node IDs are imported sequentially as edge to construct the tree, only the edges which can not form a cycle are added. In order to cover all possible tree structures, the length of the EWD encoding is defined as $2(n-1)$.

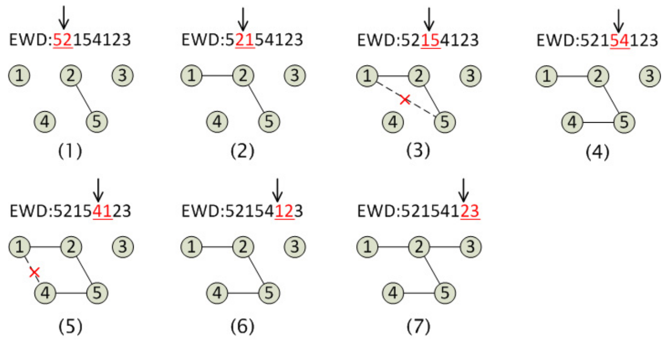

Fig. 2. Decoding process (tree construction) of EWD

Theorem 1 : The EWD encoding string with $2(n-1)$ length can represent all possible steiner trees for the involving nodes NS_m, which is a subset of NS_n, and where $|NS_n|$ is n.

Proof : Assume that the shortest length of EWD encoding string for the tree T_m is Num_m ($Num_m > 2(n-1)$), where the edge set and the node set are ES_m ($|NS_m|$=(n-2)) and NS_m ($|NS_m|$=(n-1)) respectively. If the tree T_m adds a new node through adding new edge to form a new tree T_n, then the new node set is $|NS_n| = n$ and $|ES_n| = (n-1)$. According to the property of node preserving of EWD, the NS_m and ES_m are always the subsets of the NS_n and ES_n. In addition, on the principle of the order preserving property, the length Num_n of the corresponding EWD string is more than Num_m, which means $Num_n > Num_m > 2(n-1)$. However, this conclusion violates the theorem in EWD, which denotes that a tree structure with n nodes only needs a EWD string with $2(n-1)$ lengths to be encoded. Eventually, this assumption is invalid, and the original theorem is set up.

4.4. Particle flying

Specific evolutionary operations are utilized to ensure the particles to ceaselessly fly in feasible solution space. This operator can inherit partial tree structure of attractor (current best particle), and explore the optimal position. It is noticed that the operations are mono-directional, which means that they only improve the current particle.

Depending on the comparison of the encoding content on first layer, the implementations of the evolutionary operation are different. It is caused by the unequal representing granularity for two encoding layers. If and only if encodings are different, evolutionary operation takes effect on the first layer, and additional operations are executed to improve the inheritability of tree structure from parents. Otherwise, the first layer still keeps unchangeable, and the second layer begins to specify the structure in fine granularity. The procedure of the operation is demonstrated in the following.

Algorithm 2 Particle Flying

Require: $P_i, B_j, flag, Graph(V, E)$
Ensure: O_i
  if $flag == SAME$ **then**
    $O_i^2 = EWD\_Evolution\_Operator(P_i^2, B_i^2)$;
    $O_i = P_i^1 + O_i^2$;
  else if $flag == DIFF$ **then**
    $O_i^1 = Partial\_OR\_Operator(P_i^1, B_j^1)$;
    $Tree_{P_i} = TCR\_Decoder(P_i^2)$;
    $Tree_{B_i} = TCR\_Decoder(B_i^2)$;
    $Intersection = Tree_{P_i} \cap Tree_{B_i}$;
    $Residual = (Tree_{P_i} \cup Tree_{B_i}) - (Tree_{P_i} \cap Tree_{B_i})$;
    $Edges = Check\_Endpoint(Residual, O_i^1)$;
    $Tree_{O_i} = Spanning\_Tree(Intersection, Edges)$;
    $O_i^2 = EWD\_Encode(Tree_{O_i})$;
    $O_i = O_i^1 + O_i^2$;
  end if
  return O_i;

O_i is the offspring, P_i^1 and P_i^2 are the first layer and second layer of parent respectively, B_i^1 and B_i^2 are the first layer and second layer of best solution respectively. If the first layer encoding are same, original EWD evolutionary operation is executed on second layer, which is composed by adjacent node crossover and reciprocal exchange mutation[11]. Otherwise, the first layer of the offspring O_i^1 is generated from P_i^1 and B_i^1 by partial OR operation. $Tree_{P_i}$ and $Tree_{B_i}$ are obtained from the decoding of P_i^2 and B_i^2. Only both terminals of a edge in $Residual$ selected as steiner nodes in O_i^1 can be imported into edge set $Edges$. For the purpose of better inheritance, an new tree $Tree_{O_i}$ is generated from $Intersection$ and $Edges$. Afterwards $Tree_{O_i}$ is translated to EWD encoding. Eventually, the new offspring is the combination of two layer encoding, and the partial tree structures from parents are inherited.

The time complexity of $EWD\_Evolution\_Operator$ is $O(n^2)$, where n is the number of involved nodes in steiner tree. $TCR\_Decoder$ and $Spanning\_Tree$ function are derived from Prim and Kruskal algorithm respectively, moreover, the implementation of EWD_Encode function is based on depth-first traversal, so their time complexity are all same with $EWD\_Evolution\_Operator$. Due to other operations have smaller complexity, therefore, the complexity for entire particle flying is $O(n^2)$.

4.5. Particle repair

To guarantee the validity of the solution, the particle repair based on the tree trimming operation is applied after the evolutionary operation. Since one-time tree trimming operation can only eliminate current ineligible leaf nodes, and some former relaying nodes may transform to the new ineligible leaf nodes, therefore, the trimming operation is required to be executed repeatedly until the structure become feasible. The corresponding binary for deleted nodes is changed to zero at the first layer. The identity of the deleted nodes at the second layer are directly deleted, and the residual structure is not affected.

$Theorem\,2$: Deleting the identity of leaf nodes inside the second layer encoding (EWD) equals deleting the leaf nodes in the tree structure, the residual encoding can still represent the residual tree structure.

$Proof$: Assume that n_t is the leaf node deleted at this period. Its appearance count $C(n_t)$ is more than or equals to once. According to the property of EWD decoder, when $C(n_t)$ equals to 1, which means the first time of occurrence of n_t, it has to be added to the tree. If the partial encoding sequence is $[n_i, n_t, n_{i+1}]$, only (n_i, n_t) is appended to tree, where n_i and n_{i+1} are already in the subtree, therefore, (n_i, n_{i+1}) does not lead to any adding operation. Beside the first appearance, other appearance does not lead to any edge and node appending, so (n_i, n_{i+1}) still does not invoke any adding operation. Therefore, when n_t is deleted from EWD encoding, only edge (n_i, n_t) is deleted from the tree, other edges are not influenced.

4.6. Fitness function

The fitness function is used to evaluate the performance of the solutions in evolutionary algorithm. There are two original fitness functions targeting on multi-objectives optimization. One is Pareto degree, another one is weighted sum [12]. The first method only considers the distinguishment between Pareto dominated solutions and non-dominated solutions. But the difference inside the non-dominated solutions is not concerned. The second method remedies this drawback, however, the weight allocation among multiple objectives are always not reasonable.

For the reasons given above, an adaptive hybrid function is used to evaluate the solutions. It considers integration of the advantages of both original fitness functions. The Pareto degree method is translated to penalty function, the difference of the dominated type of solutions can be distinguished. Meanwhile, in order to allocate the balanced weights in the weighted sum, the fitness are normalized by the upper and lower extreme values of each objective, and the sum of normalized values can be used to evaluate the performance further among the non-dominated solutions. If x is the current feasible solution, let Z_i^{max} and Z_i^{min} represent the maximum and minimum value of the i th objective, and their current value is $f_i(x)$ corresponding with objectives in section 3.3. The relative fitness value is updated after each evolution period. The weighted sum is denoted as

$$SUM(x) = \sum(f_i(x) - Z_i^{min})/(Z_i^{max} - Z_i^{min}) \tag{10}$$

Beside the weighted sum part, the penalty function $P(x)$ is introduced as piecewise function. If x is a non-dominated solution, $P(x) = 0$, otherwise, $P(x) = 1$. The final fitness function is the combination of these two parts, which is denoted as

$$Fintess(x) = SUM(x) + P(x) \tag{11}$$

This hybrid function remedies the limitation of traditional weighted sum function, and it is helpful to get the best solution inside Pareto front.

5. Performance Evaluation

In order to verify the feasibility and high efficiency of our heuristic approach, which is called JPSO-Double, the main metrics are observed and compared through lots of simulations. The simulations divide to three main parts. In the first part, the feasibility of JPSO-Double on random network deployment is explored. In the second part, MST-1tRNP and M-REST are selected as benchmarks to evaluate the performance of JPSO-Double, and their model are extended to support the multi-objectives defined in section 3.3. In the third part, in order to independently validate the high efficiency of encoding scheme, Prufer Number and Edge Set as the classical tree-based encoding scheme are compared with the double layer encoding.

In the experiment, the sensors are distributed within a $100m \times 100m$ area. There is only one sink, but the total number of sensors and the number of sources are variable to test the performance. The critical parameters are assumed that $E_e = 50nj/bit$, $E_{fs} = 100pj/bit/m^2$, $E_{mp} = 50pj/bit/m^2$, $E_{DA} = 5nj/bit$, and $k = 4000bit$, respectively. In addition, the transmission range of each node is 25m.

No matter the network deployment is grid, random or other cases, it can not impact the feasibility of the approach. There is the example for the random distribution in figure 3.a, in which the squared nodes, rounded nodes and triangle node indicate the source nodes, relaying nodes and sink node respectively, and its corresponding multi-objective steiner tree is depicted in figure 3.b.

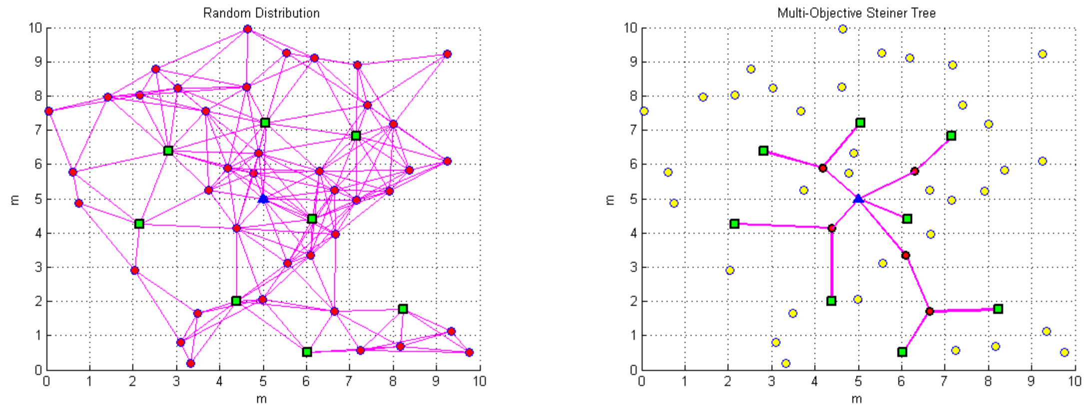

Fig. 3. (a) Random Distribution; (b) Multi-Objective Steiner Tree.

Average values of fitness obtained from 50 independent simulations with same external parameters are the metrics to measure the performance of different approximate algorithms on multi-objectives optimization.

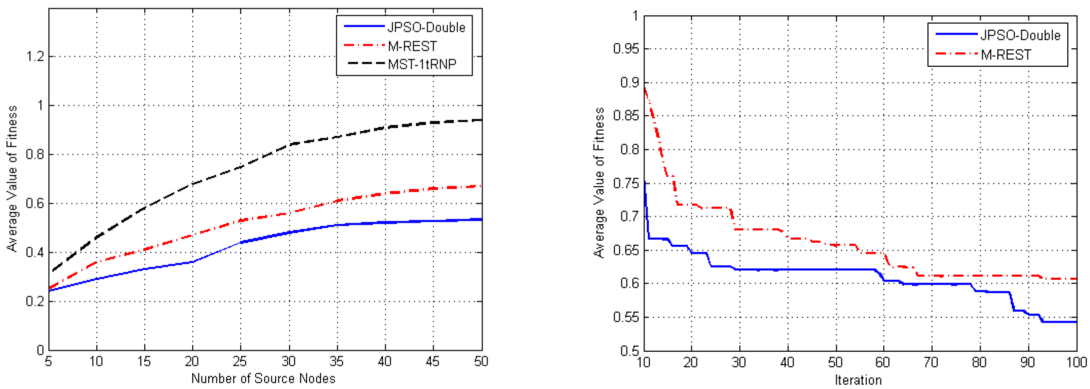

Fig. 4. (a) Comparison for different approach; (b) Learning Process.

From figure 4.a, when other parameters are stationary, as the augment of the number of source nodes, the fitness values also increase, but the growth rate gradually decrease. Because the more source nodes indicate higher cost and dimension of spanning tree, then higher fitness value is caused. Moreover, in same limited area, the more source nodes implies more chance to let source nodes become relaying nodes, and less other nodes may be selected as relaying nodes, this can slow down the growth rate of fitness value. Comparing with MST-1tRNP, the performance of JPSO-Double and M-REST are approximative and better. They are both evolutionary algorithms, and the learning process of JPSO-Double and M-REST is present to further observe the discrepancy of performance in figure 4.b. On the premise of same population for both approaches, JPSO-Double always can obtain better performance than M-REST at different iteration.

In order to independently evaluate the encoding schemes, other two tree-based encoding are combined with JPSO to solve the multi-objective optimization in our model. In figure 5, box-plot is used to reflect the quality of solutions under three encoding schemes. The indexes in JPSO-Double box have smaller fitness value than the indexes in other two boxes, such as first quartile, median, and third quartile, it indicates the solutions are closer to the theoretical optimal solution. And the shorter interquartile range means the more concentrative flying track of particles around the optimal solution. These results validate the high efficiency of double layer encoding scheme on MOSTP.

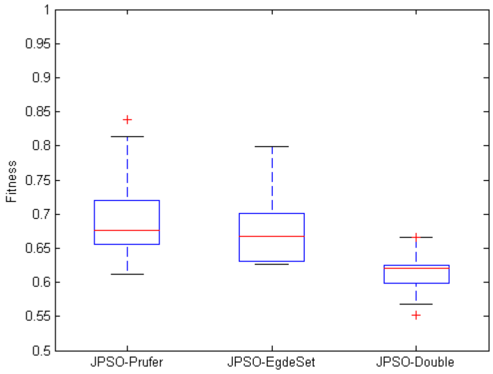

Fig. 5. Comparison for learning process

6. Conclusion and Future Work

In this paper, the modality of finding the optimal spanning tree for data aggregation in wireless sensor network is described as MOSTP. In response to the practical application requirements, four common metrics are selected as the ultimate objectives of the aggregation tree. A heuristic approach based on JPSO is presented to address this issue by utilizing the customized encoding scheme and evolutionary operation. Through the simulation results, our approach can generate the approximate optimal tree structure for MOSTP, and the performance is better than other methods. The distributed implementation of JPSO can be developed to improve the convergence time of algorithm as a future work. In addition, more other performance metrics can be imported into the multi-objective framework to achieve special requirements.

References

1. Jennifer Y, Biswanath M, Dipak G, *Wireless sensor network survey*, Computer Networks, vol.52, No. 12, p. 2292-2330, August 2008.
2. Fasolo E, Rossi M, Widmer J, Zorzi M, *In-network aggregation techniques for wireless sensor networks: a survey*, Wireless Communications, IEEE , vol.14, no.2, pp.70-87, April 2007.
3. Laukik C, Alin D, and Sanjay R, *Aggregation methods for large-scale sensor networks*, ACM Transactions on Sensor Networks (TOSN), vol. 4, no.2, pp. 1-29, March 2008.
4. Baranidharan B, Shanthi B, *An Energy Efficient Clustering Protocol Using Minimum Spanning Tree for Wireless Sensor Networks*, PDCTA 2011, Springer, pp.1-11, September 2011.
5. Upadhyayula S, Gupta SKS, *Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks*, Ad Hoc Network, vol.5, no.5, pp.626-648, July 2007.
6. Lloyd EL, Guoliang X, *Relay Node Placement in Wireless Sensor Networks*, IEEE Transactions on Computers, vol.56, no.1, pp.134-138, 2007.
7. Perez AJ, Labrador MA, Wightman PM, *A multiobjective approach to the relay placement problem in WSNs*, Wireless Communications and Networking Conference (WCNC), 2011 IEEE, pp.475-480, March 2011.
8. Heinzelman, WB, Chandrakasan, AP, Balakrishnan, H, *An application-specific protocol architecture for wireless microsensor networks*, IEEE Transactions on Wireless Communications, vol.1, no.4, pp.660-670, Oct 2002.
9. Dijun L, Xiaojun Z, Xiaobing W, Guihai C,*Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks*, INFOCOM, 2011 Proceedings IEEE, pp.1566-1574, April 2011.
10. Consoli S., Moreno Perez JA, Darby-Dowman K, Mladenovic N, *Discrete Particle Swarm Optimization for the Minimum Labelling Steiner Tree Problem*, Nature Inspired Cooperative Strategies for Optimization, vol.129, pp. 313-322, 2008.
11. Sang-Moon Soak, Corne, DW, Byung-Ha Ahn, *The edge-window-decoder representation for tree-based problems*, IEEE Transactions on Evolutionary Computation, vol.10, no.2, pp.124-144, April 2006.
12. Marler RT, Arora JS. *Survey of multi-objective optimization methods for engineering*, Structural and Multidisciplinary Optimization , vol.26, no.6 , pp. 369-395, April 2006.